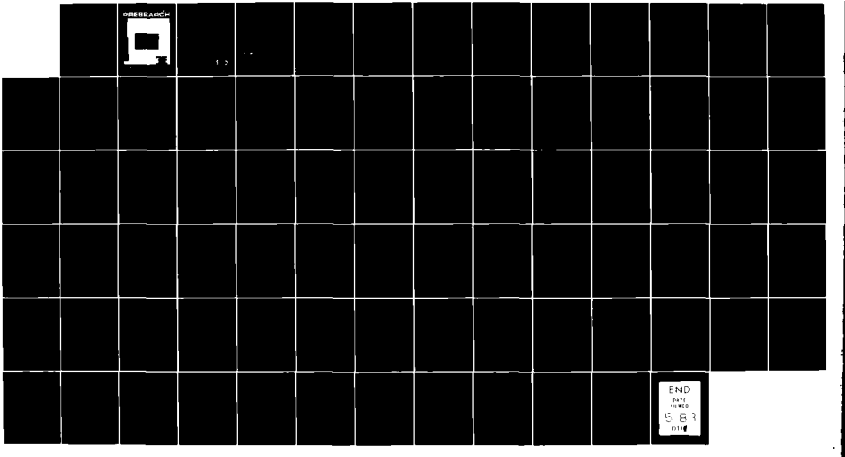
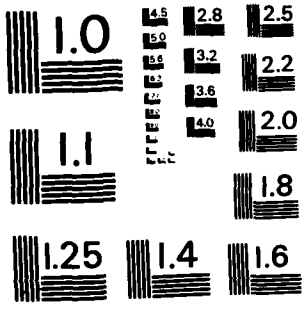
AD-A127 204 FLEET MOORING LEG DESIGN PROGRAM DOCUMENTATION VOLUME 5 - 1/1
SOURCE LISTINGS: COMPOUND LEG BASIC SOLUTION(U)
PRESEARCH INC ARLINGTON VA DEC 82 FPO-1-82-(38)
UNCLASSIFIED N62477-81-C-0025 F/G 9/2 NL





PRESEARCH

FLEET MOORING LEG
DESIGN PROGRAM DOCUMENTATION

Volume 5

SOURCE LISTINGS:
COMPOUND LEG BASIC SOLUTION

FPO-1-82-(36)

December 1982

General Distribution

PRESEARCH INCORPORATED
2361 S. JEFFERSON DAVIS HIGHWAY, ARLINGTON, VA. 22202 (703) 553-2700

PRESEARCH INCORPORATED

(1)

FLEET MOORING LEG DESIGN PROGRAM DOCUMENTATION

Volume 5

SOURCE LISTINGS: COMPOUND LEG BASIC SOLUTION

FPO-1-82-(36)

December 1982

General Distribution

Performed for
Ocean Engineering and Construction Project Office
Chesapeake Division
Naval Facilities Engineering Command
Washington, D.C. 20374
Under
Contract N62477-81-C-0025

Presearch Incorporated
2361 South Jefferson Davis Highway
Arlington, Virginia 22202

PRESEARCH INCORPORATED

FLEET MOORING LEG DESIGN PROGRAM DOCUMENTATION

Volume 5

SOURCE LISTINGS: COMPOUND LEG BASIC SOLUTION

IV. SOURCE LISTINGS

COMPOUND LEG BASIC SOLUTION

CSLACK – EDGPT, PAGES 90–164

```
e! sys final/f2for/cslack for##
      subroutine CSLACK
*************************************************************************
      implicit integer*2 (i)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      integer*2 iuks
      equivalence (uz(3),iuks)
```

```
      double precision pi ,halfpi ,degrad ,raddeg ,zero ,one ,half
      integer*2 izero ,ione ,itwo
      common /VCONST/ pi ,halfpi ,degrad ,raddeg ,zero ,one ,half ,
     &  izero ,ione ,itwo

      integer*2 iscopa ,iscopb ,itana ,itanb ,it ,is
      double precision epsy ,gamma ,se
      common /VCMPD/ epsy ,gamma ,se ,iscopa ,iscopb ,itana ,itanb ,it ,is

      integer*2 itold
      double precision ss0 ,dten0 ,ss1 ,dten1 ,ss2 ,dten2 ,slp0 ,sa0 ,smin(2)
      common /VEQUAL/ ss0 ,dten0 ,ss1 ,dten1 ,ss2 ,dten2 ,slp0 ,sa0 ,smin ,
     &  itold
      equivalence (smin(1) ,samin) ,(smin(2) ,sbmin)

      double precision sa ,sb ,ca ,cb ,vc0a(6) ,vc0b(6) ,
     &  eex0 ,eez0 ,eey0 ,a0 ,b0 ,phia0 ,phib0
      integer*2 icase
      common /VSPID/ sa ,sb ,ca ,cb ,vc0a ,vc0b ,
     &  eex0 ,eez0 ,eey0 ,a0 ,b0 ,phia0 ,phib0 ,
     &  icase

      double precision snphih ,csphih ,snafh ,csafh ,tnafh ,scafh ,dsnph
      common /VHDIR/ snphih ,csphih ,snafh ,csafh ,tnafh ,scafh ,dsnph

      double precision hinafh ,hw4 ,w4h ,s4w4h ,c3h
      common /VHVEC/ hinafh ,hw4 ,w4h ,s4w4h ,c3h

      double precision epsxz ,xziru(2) ,xzbas(2) ,hbas(2) ,scrat1(10)
      common /VCSSXZ/epsxz ,xziru ,xzbas ,hbas ,scrat1
      double precision xiru ,ziru ,xbas ,zbas ,hbasx ,hbasz
      equivalence (xziru(1) ,xiru) ,(xziru(2) ,ziru) ,
     &  (xzbas(1) ,xbas) ,(xzbas(2) ,zbas) ,
     &  (hbas(1) ,hbasx) ,(hbas(2) ,hbasz)

      integer*2 itant
      double precision a ,b ,snphi ,tnafa ,tnafb ,
     &  seca7 ,seca8 ,ut ,st ,ykt ,zkt ,eex ,eez ,eey ,ybuoy
      common /VCSSHP/ a ,b ,snphi ,tnafa ,tnafb ,
     &  seca7 ,seca8 ,ut ,st ,ykt ,zkt ,eex ,eez ,eey ,ybuoy ,itant

      integer*2 ivs
      double precision v0 ,v1 ,v2 ,f0 ,f1 ,f2 ,f ,eps
```

```
      common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
      double precision varray(3),farray(3)
      equivalence (v0,varray),(f0,farray)

      integer*2 ilh0,ilh1,ilh2,il
      double precision lh0,lh1,lh2,ce
      common /VSCOIL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
      integer*2 ilh(3)
      double precision lh(3)
      equivalence (ilh,ilh0),(lh,lh0)

      double precision xred
      integer*2 isidf,nerra,nerrb
      common /VSTAB/ xred,isidf,nerra,nerrb
*******************************************************************************
* BEGIN EXECUTABLE CODE
*******************************************************************************
      call ovlink('CPREP0 ')
      if (ileg eq 3) call CPREP1
      goto (1000,2000,3000), iuks

 1000 continue
      call CPREP2
      call CPREP3
      if (ileg ne 3) goto 1200
      call ovlink('CSSHP ')
      goto 1500
 1200 continue
      call ovlink('CSEHP ',0)
 1500 continue
      goto 5000

 2000 continue
      call CPREP2
      if (ileg ne 3) goto 2200
      call ovlink('CSSPR ',0)
      goto 2500
 2200 continue
      call ovlink('CSEPR ',0)
 2500 continue
      goto 5000

 3000 continue
```

```
      if (ileg ne 3) goto 3200
      call ovlink('CSSXZ1  ')
      call ovlink('CSSXZ2  ')
      goto 3500
 3200 continue
      call ovlink('CSEXZ1  ')
      call ovlink('CSEXZ2  ')
 3500 continue

 5000 continue
      return
      end
*
```

93

```
e! sys final/:2for/cprep0 for**
      subroutine CPREP0
*************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VCL0B/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      integer*2 nc(2)
      equivalence (nca,nc)
```

94

```
      integer*2 uz1,uz2
      equivalence (uz(1),uz1),(uz(2),uz2)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      integer*2 iscopa,iscopb,ifana,ifanb,if,is
      double precision epsy,gamma,se
      common /VCMPD/ epsy,gamma,se,iscopa,iscopb,ifana,ifanb,if,is

      integer*2 ifold
      double precision ss0,dten0,ss1,dten1,ss2,dten2,slp0,sa0,smin(2)
      common /VEQUAL/ ss0,dten0,ss1,dten1,ss2,dten2,slp0,sa0,smin,
     &  ifold
      equivalence (smin(1),samin),(smin(2),sbmin)

      integer*2 il,i,j,n
*************************************************************************
      ifana=2*nca+17
      ifanb=2*ncb+17
      iscopa=3*nca
      iscopb=3*ncb+25
      se=z(iscopa)+z(iscopb)

      yk=cz*halfd
      do 20 il=1,2
      smn=(do-yk)-s4
      n=nc(il)
      do 10 i=1,n
      if (i eq 1) goto 10
      j=25*(il-1)+3*(i-1)
      smn=smn-z(j)
   10 continue
      smin(il)=dmax1(smn,zero)
      yk= -yk
   20 continue

      slp0=dmax1(dmin1(slp,z(iscopa)-samin),sbmin-z(iscopb))
```

```
sa0-z(,scopa)

return
end
```

96

```
e! sys final/f2for/cprep1 for##
      subroutine CPREP1
*************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),s1a),(za(4),w1a),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
```

```
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
     &  eex0,eez0,eey0,a0,b0,phia0,phib0
      integer*2 icase
      common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
     &  eex0,eez0,eey0,a0,b0,phia0,phib0,
     &  icase

      equivalence (czsq1,eez0),(ddsq,eta,eex0),(aasq,eey0),
     & (aambb,eex0sq,b0sq,phib0),(a0sq,phia0)
*************************************************************************
      call SUMSC(nca,za,sa,ca)
      call SUMSC(ncb,zb,sb,cb)
      call VCRIT0(nca,za,vc0a)
      call VCRIT0(ncb,zb,vc0b)
      czsq1=cz*cz+one
      ddsq=dsq*czsq1
      icase=1
      if ((sa-sb)**2 gt ddsq) goto 100
      icase=2
      aasq=sa*sa
      aambb=aasq-sb*sb
      eta=dsqrt((4.0d0*aasq*ddsq-(ddsq+aambb)**2)/(cx*cx+czsq1))
      eez0= -(aambb+cz*cx*eta)/(twod*czsq1)
      eex0=eta/twod
      eey0=cx*eex0+cz*eez0
      eex0sq=eex0*eex0
      a0sq=eex0sq+(eez0-halfd)**2
      b0sq=eex0sq+(eez0+halfd)**2
      a0=dsqrt(a0sq)
      b0=dsqrt(b0sq)
      call PHIAB(a0,b0,a0sq-b0sq,dsq,twod,phia0,phib0)
  100 continue
      return
      end
*
```

```
e! sys final/12for/sumsc for##
      subroutine SUMSC(nc,z,s,c)
*******************************************************************************
      implicit double precision (a-z)

      integer*2 nc
      double precision z(25),s,c
*******************************************************************************
      s=z(3)
      c=s*z(4)
      if (nc eq 1) goto 100
      s=s+z(6)
      c=c+z(5)+z(6)*z(7)
      if (nc eq 2) goto 100
      s=s+z(9)
      c=c+z(8)+z(9)*z(10)
  100 continue
      return
      end
*
```

```
e! sys final/!2for/cprep2 for##
      subroutine CPREP2
*************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),s1a),(za(4),w1a),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frc1,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frc1),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
```

```
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision tnaf,phif
      common /VOFLR/ tnaf,phif

      double precision snphih,csphih,snafh,csafh,tnafh,scafh,dsnph
      common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh,dsnph
*****************************************************************************
      csphih=dcos(phih)
      snphih=dsin(phih)
      tnafh=dcos(phih-phif)*tnaf
      scafh=SECNT(tnafh)
      snafh=tnafh/scafh
      csafh=one/scafh
      dsnph=d*snphih
      return
      end
*
```

```
e! sys final/f2for/cprep3 for##
      subroutine CPREP3
*************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,xla,x2a,x3a,yla,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),xla),(za(14),x2a),(za(15),x3a),
     &  (za(16),yla),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),xlb),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),ylb),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision snphih,csphih,snafh,csafh,tnafh,scafh
      common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh
```

```
      double precision hinafh,hw4,w4h,s4w4h,c3h
      common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h
*****************************************************************************
      hinafh=h*inafh
      hw4=h/w4
      w4h=w4/h
      s4w4h=s4*w4h
      c3h=c3/h
      return
      end
*
```

```
et sys final/t2for/csshp for##
      subroutine CSSHP
*************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
```

104

```
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision inaf,phif
      common /VOFLR/ inaf,phif

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      integer*2 iscopa,iscopb,iianc,iianb,ie
      double precision epsy,gamma,se
      common /VCMPD/ epsy,gamma,se,iscopa,iscopb,iianc,iianb,ie

      double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
     &  eex0,eez0,eey0,a0,b0,phia0,phib0
      integer*2 icase
      common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
     &  eex0,eez0,eey0,a0,b0,phia0,phib0,
     &  icase

      double precision snphih,csphih,snafh,csafh,inafh,scafh
      common /VHDIR/ snphih,csphih,snafh,csafh,inafh,scafh

      double precision hinafh,hw4,w4h,s4w4h,c3h
      common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h

      integer*2 iiani
      double precision a,b,snphi,inafa,inafb,
     &  seca7,seca8,ui,si,yki,zki,eex,eez,eey,ybuoy
      common /VCSSHP/ a,b,snphi,inafa,inafb,
     &  seca7,seca8,ui,si,yki,zki,eex,eez,eey,ybuoy,iiani

      integer*2 ivs
      double precision v0,v1,v2,f0,f1,f2,f,eps
      common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs

      double precision xred
      integer*2 isidf,nerra,nerrb
      common /VSTAB/ xred,isidf,nerra,nerrb

      integer*2 ieb,nw,ntest,nerr
*****************************************************************************
```

```
      epsy=do*1 0d-10
      ibrnch=1
      ieb=0

      nw=0
      if (nwa eq 0  and  nwb eq 0) goto 500
      nw=1
      if (nwa eq 1  and  nwb eq 1) goto 4000
      if (nwb eq 1) ibrnch=2
      goto 3000

*****************************************************************************
* Determine the number of branches under tension
* when junction lies on ocean floor
*****************************************************************************
  500 continue
*****************************************************************************
* If branch lengths differ by more than distance between anchors,
* then one branch is under tension
*****************************************************************************
      if (icase eq 1) goto 2000
*****************************************************************************
* If load is directed outside angle formed by branch extensions,
* then one branch is under tension
*****************************************************************************
      if (phih lt phia0  or  phih gt phib0) goto 2000

*****************************************************************************
* Both branches are under tension if junction lies on ocean floor
* Assume this to be the case, with branches forming a triangle
* Calculate buoy elevation when junction is just lifted off floor
*****************************************************************************
      phia=phia0
      phib=phib0
      call HSPLIT
      tana7=(c3+ha*tnafa+hb*tnafb)/h
      call TRISR
      ybuoy=eey0+y4
      if (ybuoy lt do) goto 1200
*****************************************************************************
* Computed buoy elevation is no less than water level,
* therefore junction lies on ocean floor
* Calculate riser displacements and angles directly
```

```
*************************************************************************
      call SRISR(eey0,one)
      call TSPLIT
      call JUNCT(tnafa,tnafb,0)
      eex=eex0
      eez=eez0
      isol=1
      goto 5000
*************************************************************************
* Computed buoy elevation is less than water level,
* therefore junction lies above ocean floor
* Test each branch for plane solution before searching for solution
* in three dimensions, shorter branch is first to be tested
*************************************************************************
 1200 continue
      if (sa gt sb) ibrnch=2
      goto 3000

*************************************************************************
* One branch is under tension if junction lies on ocean floor
* This is true for one for two reasons, as indicated by index 'icase'
*     1 if branch lengths differ by more than distance between anchors
*     2 if load is directed outside angle formed by branch extensions
* Assume that junction lies on ocean floor
*************************************************************************
* Determine index 'ibrnch' of branch under tension   1 for A,  2 for B
*************************************************************************
 2000 continue
      if (icase ne 1) goto 2010
      if (sa gt sb) ibrnch=2
      goto 2015
 2010 continue
      if (phih gt phib) ibrnch=2
 2015 continue
*************************************************************************
* Set parameters for branch under tension
* Calculate buoy elevation when junction is just lifted off ocean floor
*************************************************************************
      call EBUOY
      if (ybuoy lt do) goto 2200
*************************************************************************
* Computed buoy elevation is no less than water level,
* therefore junction lies on ocean floor
```

107

```
* Calculate riser displacements and angles directly
*******************************************************************************
      call SRISR(eey,one)
      call JTEN(ta)
      tb=zero
      if (ibrnch eq 1) goto 2110
      tb=ta
      ta=zero
 2110 continue
      gamma=pi
      eex=sf*csafh
      eez=zkf+eex*snphih
      eex=eex*csphih
      isol=2
      goto 5000
*******************************************************************************
* Computed buoy elevation is less than water level,
* therefore junction lies above ocean floor
* Test each branch for plane solution before searching for solution
* in three dimensions, beginning with branch currently assumed to be
* under tension
*******************************************************************************
 2200 continue
      ieb=1

*******************************************************************************
* Possibility of junction on floor has been eliminated
* For each branch, find solution in vertical plane of loading force,
* and compute straight-line distances for other branch
*******************************************************************************
 3000 continue
      ntest=1
 3010 continue
      if (ieb eq 1) goto 3100
      call EBUOY
      if (nw eq 0  and  ybuoy gt dol) goto 3500
 3100 continue
      if (ibrnch eq 2) goto 3150
      call SCOIL(nca,za,vc0a,ca,nwa,ncb,zb,vc0b,sb,nerr)
      goto 3200
 3150 continue
      call SCOIL(ncb,zb,vc0b,cb,nwb,nca,za,vc0a,sa,nerr)
 3200 continue
```

```
      if (nerr eq 0  and  coil ge zero) goto 3600
*************************************************************************
* No possibility of plane solution in current tension branch
* Either depth is insufficient (from subroutine EBUOY),
* or other branch is too short (from subroutine SCOIL)
*************************************************************************
 3500 continue
      if (ntest eq 2  or  nw eq 1) goto 4000
      ibrnch=3-ibrnch
      ntest=2
      ieb=0
      goto 3010
*************************************************************************
* Solution in plane of loading force is consistent with length of
*  other branch
*************************************************************************
 3600 continue
      isol=3
      goto 5000

*************************************************************************
* Solution must be three-dimensional with junction above ocean floor,
* solve by iteration over sides a,b of horizontal triangle
*************************************************************************
 4000 continue
      call STEFAB
      call JUNCT(za(ita),zb(itb),1)
      eex=xa*dcos(phia)
      eez=halfd+xa*dsin(phia)
      isol=4

*************************************************************************
* Final computations for all solution types
*************************************************************************
 5000 continue
      gamma=dexp(gamma*frct)
      xtot=eex+x4*csphih
      ztot=eez+x4*snphih
      rtot=csphih*xtot+snphih*ztot
      return
      end
*
```

```
e! sys final/!2for/ph!ab for##
      subroutine PHIAB(a,b,aambb,dsq,!wod,ph!a,ph!b)
*************************************************************************
      implicit double precision (a-z)

      double precision a,b,aambb,dsq,!wod,ph!a,ph!b

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 !zero,!one,!two
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &   !zero,!one,!two
*************************************************************************
      ph!a=dacos((dsq+aambb)/(a*!wod))-halfpi
      ph!b=halfpi-dacos((dsq-aambb)/(b*!wod))
      return
      end
*
```

```
e! sys final/12for/hsplit for##
      subroutine HSPLIT
*************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision inaf,phif
      common /VOFLR/ inaf,phif
```

```
      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
      common /VCSSHP/ a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
*************************************************************************
      snphi=dsin(phib-phia)
      ha=h*dsin(phib-phih)/snphi
      hb=h*dsin(phih-phia)/snphi
      tnafa=dcos(phia-phif)*tnaf
      tnafb=dcos(phib-phif)*tnaf
      return
      end
*
```

112

```
ei sys final/t2for/trisr for##
      subroutine TRISR
*************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,c1b,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),c1b),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision hinafh,hw4,w4h,s4w4h,c3h
      common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h
```

```
      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
      common /VCSSHP/ a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
*****************************************************************************
      tana8=tana7+s4w4h
      call SCA7A8
      y4=hw4*(seca8-seca7)
      return
      end
*
```

114

```
e! sys final/!2for/sca7a8 for!!
      subroutine SCA7A8
*****************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ia,ib
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ia,ib,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
```

```
      common /VCSSHP/ a,b,snphi,tnafa,tnafb,
     &   seca7,seca8,ui,si,yki,zki,eex,eez,eey,ybuoy
*****************************************************************************
      seca7=SECNT(iano7)
      seca8=SECNT(iano8)
      return
      end
*
```

```
e! sys final/f2for/srisr for##
      subroutine SRISR(ey,lfact)
*************************************************************************
      implicit double precision (a-z)

      double precision ey,lfact

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
```

```
      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision snphih,csphih,snafh,csafh,tnafh,scafh
      common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh

      double precision hinafh,hw4,w4h,s4w4h,c3h
      common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h
*******************************************************************************
      y4=do-ey
      l-LENS(y4,csafh,snafh,s4,w4,h)*lfact
      if (l gt zero) goto 20
      l=zero
      tana7=TAN1(s4w4h,y4*w4h)
      goto 50
   20 continue
      tana7=tnafh
   50 continue
      tana8=tana7+(s4-l)*w4h
      call SCA7A8
      call X4CALC
      x4=x4+l*csafh
      return
      end
*
```

```
et sys final/f2for/tanl for$$
      function TANL(dt,ds)
*************************************************************************
* Computes tangent of the algebraically smaller of two angles,
* given the differences between their tangents and secants
*************************************************************************
      implicit double precision (a-z)

      double precision tanl,dt,ds

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo
*************************************************************************
* tanl = tangent of smaller angle
* dt = tangent of larger angle - tangent of smaller angle
* ds = secant of larger angle - secant of smaller angle
*************************************************************************
      tanl=half*(ds*dsqrt(one+4.0d0/(dt*dt-ds*ds))-dt)
      return
      end
*
```

```
et sys final/f2for/x4calc for##
      subroutine X4CALC
*************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision hinafh,hw4,w4h,s4w4h,c3h
      common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h
```

```
      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
      common /VCSSHP/ a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
*******************************************************************************
      x1=hw1*dlog((tana8+seca8)/(tana7+seca7))
      return
      end
*
```

```
ef sys final/12for/tsplit for##
      subroutine TSPLIT
*************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),s1a),(za(4),w1a),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      double precision b,sinb,cosb,tanb,secb
      equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)
```

122

```
      integer*2 uz1,uz2,uz3,uz4,uz5
      equivalence (uz(1),uz1),(uz(2),uz2),(uz(3),uz3),(uz(4),uz4),
     &  (uz(5),uz5)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
     &  eex0,eez0,eey0,a0,b0,phia0,phib0
      integer*2 icase
      common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
     &  eex0,eez0,eey0,a0,b0,phia0,phib0,
     &  icase

      double precision snphih,csphih,snafh,csafh,tnafh,scafh
      common /VHOIR/ snphih,csphih,snafh,csafh,tnafh,scafh

      equivalence (czsql,ddsq,phibb,eex),(fx,phihh,eez),
     &  (fz,dd,snphii,eey),(phiaa,gamma)
*****************************************************************************
      czsql=cz*cz+one
      fx=czsql*csphih-cx*cz*snphih+cx*tnafh
      fz=snphih+cz*tnafh
      if (fx ne zero) goto 20
      if (fz lt zero) goto 10
      phihh=halfpi
      goto 15
   10 continue
      phihh= -halfpi
   15 continue
      goto 50
   20 continue
      phihh=datan(dsqrt(cx*cx+czsql)*fz/fx)
   50 continue
      ddsq=dsq*czsql
      dd=dsqrt(ddsq)
      call PHIAB(sa,sb,sa*sa-sb*sb,ddsq,dd+dd,phiaa,phibb)
      snphii=dsin(phibb-phiaa)
```

123

```
      call JTEN(ienj)
      ia=tenj*dsin(phibb-phihh)/snphi
      ib=tenj*dsin(phihh-phiaa)/snphi
      return
      end
*
```

124

```
et sys final/t2for/jten for##
      subroutine JTEN(tenj)
*************************************************************************
      implicit double precision (a-z)

      double precision tenj

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      double precision b,sinb,cosb,tanb,secb
```

125

```
      equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)
      integer*2 uz1,uz2,uz3,uz4,uz5
      equivalence (uz(1),uz1),(uz(2),uz2),(uz(3),uz3),(uz(4),uz4),
     &  (uz(5),uz5)

      double precision snphih,csphih,snafh,csafh,inafh,scafh
      common /VHDIR/ snphih,csphih,snafh,csafh,inafh,scafh
*****************************************************************************
      tenj=h*SECNT(iana7)*dcos(datan(iana7)-datan(inafh))
     &  -(c3+1)*w4)*snafh
      return
      end
*
```

126

```
ei sys final/i2for/junct for##
      subroutine JUNCT(tna,tnb,index)
*************************************************************************
      implicit double precision (a-z)

      integer*2 index
      double precision tna,tnb

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
```

127

```
      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      integer*2 iscopa,iscopb,itana,itanb,ie
      double precision epsy,gamma,se
      common /VCMPD/ epsy,gamma,se,iscopa,iscopb,itana,itanb,ie

      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,uf,st,ykt,zkt,eex,eez,eey,ybuoy
      common /VCSSHP/ a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,uf,st,ykt,zkt,eex,eez,eey,ybuoy

      equivalence (csphi,gamma),(sca,eex),(scb,eez)
*****************************************************************************
      sca=SECNT(ina)
      scb=SECNT(inb)
      csphi=dsqrt(one-snphi*snphi)
      if (phib-phia gt halfpi) csphi= -csphi
      gamma=pi-dacos((ina*inb+csphi)/(sca*scb))
      if (index ne ione) goto 100
      ia=ha*sca
      ib=hb*scb
  100 continue
      return
      end
*
```

128

```
et sys final/t2for/ebuoy for##
      subroutine EBUOY
*************************************************************************
* Assigns values to parameters ut,st,ykt,zkt for branch assumed to be
* under tension   Assumes junction to be just off ocean floor
* Computes junction y-coord, vertical riser displacement, buoy y-coord
*
* ibrnch = index of tension branch   1 for A,  2 for B
* ut = unit factor for tension branch   1 for A,  -1 for B
* st = length of tension branch
* halhb = load on tension branch
* ykt = y-coord of tension branch anchor
* zkt = z-coord of tension branch anchor
* eey = y-coord of junction, in general, of point on floor
*       directly beneath junction
* ybuoy = y-coord of buoy
*************************************************************************
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,a1b,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),a1b,vb),
     &  (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
```

```
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
     &  eex0,eez0,eey0,a0,b0,phia0,phib0
      integer*2 icase
      common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
     &  eex0,eez0,eey0,a0,b0,phia0,phib0,
     &  icase

      double precision snphih,csphih,snafh,csafh,tnafh,scafh
      common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh

      double precision htnafh,hw4,w4h,s4w4h,c3h
      common /VHVEC/ htnafh,hw4,w4h,s4w4h,c3h

      integer*2 iscopa,iscopb,itana,itanb,ie
      double precision epsy,gamma,se
      common /VCMPD/ epsy,gamma,se,iscopa,iscopb,itana,itanb,ie

      integer*2 itant
      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
      common /VCSSHP/ a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy,itant
*****************************************************************************
```

```
      if (ibrnch ne 1) goto 20
      u1=one
      s1=sa
      ha=h
      ilan1=ilana
      goto 50
   20 continue
      u1= -one
      s1=sb
      hb=h
      ilan1=ilanb
   50 continue
      zk1=u1*halfd
      yk1=cz*zk1
      eey=yk1+s1*snafh
      tana7=tnafh+c3h
      call TRISR
      ybuoy=eey+y4
      return
      end
*
```

131

```
e! sys final/12for/scoil for##
      subroutine SCOIL(nct,zt,vc0t,ct,nwt,ncc,zc,vc0c,sc,nerr)
*************************************************************************
      implicit integer*2 (n)
      implicit double precision (a-z)

      integer*2 nct,nwt,ncc,nerr
      double precision zt(25),vc0t(6),ct,zc(25),vc0c(6),sc

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VCLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
```

```
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      integer*2 iscopa,iscopb,ifana,ifanb,ie
      double precision epsy,gamma,se
      common /VCMPD/ epsy,gamma,se,iscopa,iscopb,ifana,ifanb,ie

      double precision snphih,csphih,snafh,csafh,tnafh,scafh
      common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh

      double precision hinafh,hw4,w4h,s4w4h,c3h
      common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h

      integer*2 ifant
      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
      common /VCSSHP/ a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy,ifant

      integer*2 ivs
      double precision v0,v1,v2,f0,f1,f2,f,eps
      common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs

      integer*2 ilh0,ilh1,ilh2,il
      double precision lh0,lh1,lh2,ce
      common /VSCOIL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
      integer*2 ilh(3)
      double precision lh(3)
      equivalence (ilh,ilh0),(lh,lh0)

      equivalence (kte,coil),(kcesq,tanaj,gamma)
*************************************************************************
*     write(10,*) 'SCOIL',ibrnch,ybuoy
      f=do
      ivs=0
      call XSECV(inct,zt,vc0t,st,ct,nwt,ncc,zc,vc0c,
     &  snafh,csafh,tnafh,scafh,4,nerr)
      if (nerr ne 0) goto 5000
```

133

```
2000 continue
     call X4CALC
     kie=zi(11)
     eex=kie*csphih
     eez=zki+kie*snphih
     kcesq=eex*eex+(zki+eez)**2
     coil=sc-dsqrt(kcesq+(yki+yki+kie*tnafh)**2)-lh2
     ianaj=zi(iiani)
     ia=h*SECNT(ianaj)
     ib=ce
     if (ibrnch eq 1) goto 2110
     ib=ia
     ia=ce
2110 continue
     gamma=datan(ianaj)+halfpi

5000 continue
     return
     end
*
```

134

```
et sys final/12for/xsecv for;1
      subroutine XSECV(nct,zt,vc0t,stp,ct,nwt,ncc,zc,vc0c,
     &  sinaf,cosaf,tanaf,secaf,iftyp,nerrf)
*************************************************************************
      implicit integer*2 (n)
      implicit double precision (a-z)

      integer*2 nct,nwt,ncc,iftyp,nerrf
      double precision zt(25),vc0t(6),stp,ct,zc(25),vc0c(6),
     &  sinaf,cosaf,tanaf,secaf

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VCLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
```

```
&  (z(59),tana7),(z(60),tana8),(z(61),l),
&  (z(62),h),(z(63),phih),
&  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

 double precision pi,halfpi,degrad,raddeg,zero,one,half
 integer*2 izero,ione,itwo
 common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
&  izero,ione,itwo

 double precision snphih,csphih,snafh,csafh,tnafh,scafh
 common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh

 double precision hinafh,hw4,w4h,s4w4h,c3h
 common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h

 integer*2 itant
 double precision a,b,snphi,tnafa,tnafb,
&  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,vbuoy
 common /VCSSHP/ a,b,snphi,tnafa,tnafb,
&  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,vbuoy,itant

 integer*2 ivs
 double precision v0,v1,v2,f0,f1,f2,f,eps
 common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
 double precision varray(3),farray(3)
 equivalence (v0,varray),(f0,farray)

 double precision fred
 integer*2 isidf,nerra,nerrb
 common /VSTAB/ fred,isidf,nerra,nerrb

 integer*2 ilh0,ilh1,ilh2,il
 double precision lh0,lh1,lh2,ce
 common /VSCOIL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
 integer*2 ilh(3)
 double precision lh(3)
 equivalence (ilh,ilh0),(lh,lh0)

 integer*2 nerr,nit,i,j,n,ivint,ndiv,is,in

 equivalence (ht,ta),(htanaf,tb),(htenth,gamma),(dv,eex),
&  (vbase,eez),(fbase,eey),(absfb,rtot),(flim,xtot),(len,ztot),
&  (ctht,coil)
```

```
      equivalence (len,delv,v0sav),(clht,delv0,vlow,vlsav),
     &  (delvl,vhigh,f0sav),(rat,hterm,flsav),(ratmax,rat0),
     &  (is,in,ivint)
*****************************************************************************
      nerrf=1
      ht=zt(1)
*     write(10,*) 'XSECV',f,ht,iftyp
      htanaf=ht*tanaf
      small=1.0d-10
      eps=f*small
      if (ivs.eq.0) goto 1100
      call SHIFT(0,1)
      call SHIFT(1,2)
      goto 5500
 1100 continue
      len=1.0d1
      htenth=ht/len
      dv=ct*1.0d-3

      n=2*nct
      do 1200 i=2,n
      if (vc0t(i).lt.vc0t(i-1)) goto 1200
      j=i-1
      goto 1210
 1200 continue
      j=n
 1210 continue
      vbase=vc0t(j)+htanaf
      v0=vbase
      call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
     &  htanaf,0,iftyp,nerr)
*****************************************************************************
*     if (f0.ne.zero) goto 1250
*     call SHIFT(2,0)
*     nerrf=0
*     goto 6000
*1250 continue
*****************************************************************************
      fbase=f0
      absfb=dabs(fbase)
      fred=absfb
      flim=-f
```

```
      if (iftyp ne 1) flim=flim+stp
      if (iftyp eq 4) flim=flim+ykt+s4
*****************************************************************************
*     if (iftyp ne 2) goto 1500
*     len=LENS(f,cosaf,sinaf,stp,ct/stp,ht)
*     if (len gt zero) goto 1320
*     cthi=ct/ht
*     v0=ht*TAN((cthi,f*cthi/stp)+ct
*     goto 1350
*1320 continue
*     v0=ct*(one-len/stp)+htanaf
*1350 continue
*     v0=dmax1(v0,vbase)
*     call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
*    &  htanaf,0,2,nerr)
*     v1=v0+ten*dv
*     call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
*    &  htanaf,1,2,nerr)
*     goto 5500
*1500 continue
*****************************************************************************
      if (fbase*flim ge zero) goto 3000

*     write(10,*) '2'
      delv1=htenth
 2100 continue
      v1=ct+htanaf+delv1
      call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
     &  htanaf,1,iftyp,nerr)
      if (f0*f1 lt zero) goto 4000
      call SHIFT(0,1)
      delv1=delv1*ten
      goto 2100

 3000 continue
*     write(10,*) '3'
      nit=0
      delv1=dv
      ratmax=ct/(stp*small)
      v1=vbase+delv1
      goto 3150
 3100 continue
      rat=(v0-vbase)/(f0-fbase)
```

```
      if (dabs(rat) gt ratmax) goto 3500
      vl=vbase-fbase*rat
 3150 continue
      call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
     & htanaf,1,iftyp,nerr)
      fred=dmin1(fred,dabs(fl))
      if (f0*fl lt zero) goto 4000
      if (dabs(fl) ge absfb) goto 3500
      delvl=vl-v0
      if (nit le 1) goto 3160
      if (dabs(one-delv*delvl/delv0**2) lt one/ten) goto 5500
 3160 continue
      call SHIFT(0,1)
      delv=delv0
      delv0=delvl
      nit=nit+1
      goto 3100

 3500 continue
*     write(10,*) '7'
      if (nit eq 0  and  iftyp eq 1  and  nwt eq 0) goto 6000
      vlow=vbase
      hterm=htenth
      do 3600 ivint=1,4
      ndiv=7-ivint
      vhigh=ct+htanaf+hterm
      delv=vhigh-vlow
      n=1
      do 3550 i=1,ndiv
      vl=vlow+half*delv
      do 3540 j=1,n
      call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
     & htanaf,1,iftyp,nerr)
      if (f0*fl lt zero) goto 4000
      fred=dmin1(fred,dabs(fl))
      vl=vl+delv
 3540 continue
      n=n+n
      delv=half*delv
 3550 continue
*******************************************************************************
*     vl=vhigh
*     call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
```

```
*     &  hianaf,1,iftyp,nerr)
*       if (f0*f1 lt zero) goto 4000
*       fred=dmin1(fred,dabs(f1))
*************************************************************************
      vlow=vhigh
      hterm=hterm*ten
 3600 continue
      goto 6000

 4000 continue
*     write(10,*) '4'
      nit=1
 4100 continue
      if (iftyp ne 4) goto 4200
      if (dabs(v0-v1) gt dv  or  dabs(ra10) gt dv/sip) goto 4200
      j=ihi-ih0
      if (j*j ne 1) goto 4200
      j=(3-j)/2
      n=jhi(j)
      f=zero
      is=3*ncc
      do 4120 i=1,n
      f=f+zc(is)
      is=is-3
 4120 continue
      v0sav=v0
      v1sav=v1
      f0sav=f0
      f1sav=f1
      f0=ih0-f
      f1=ih1-f
      call SECVIT(nct,zt,vc0t,ncc,zc,vc0c,snafh,csafh,tnafh,
     &  scafh,hinafh,3,nerr)
      if (nerr ne 0) goto 4140
      call SRISR(ykt+zt(12),zero)
      ce=h*(tana7-zt(iant))-c3
      in=2*(ncc-j)
      if (ce lt vc0c(in+1)  or  ce gt vc0c(in)) goto 4140
      nerrf=0
      goto 6000
 4140 continue
      f=do
      v0=v0sav
```

```
      v1=v1sav
      f0=f0sav
      f1=f1sav
      ilh(j)=n
      j=3-j
      ilh(j)=n+1
 4200 continue
      v2=half*(v0+v1)
      if (nit gt 3) v2=v1-f1*rat0
      call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
     &  htanaf,2,iftyp,nerr)
      i=1
      if (f2*f0 gt zero) i=0
      call SHIFT(i,2)
      rat1=(v1-v0)/(f1-f0)
      if (dabs(f0)+dabs(f1) lt eps*ten) goto 5500
      if (nit ge 5  and  dabs(rat0/(rat0-rat1)) gt ten  and
     &  (iftyp ne 4  or  ilh0 eq ilh1)) goto 5500
      rat0=rat1
      nit=nit+1
      goto 4100

 5500 continue
      call SECVIT(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,
     &  secaf,htanaf,iftyp,nerrf)

 6000 continue
      if (fbase lt zero) fred= -fred
      return
      end
*
```

```
ei sys final/i2for/shift for##
      subroutine SHIFT(i,j)
*************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 i,j

      integer*2 ivs
      double precision v0,v1,v2,f0,f1,f2,f,eps
      common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
      double precision varray(3),farray(3)
      equivalence (v0,varray),(f0,farray)

      integer*2 ilh0,ilh1,ilh2,il
      double precision lh0,lh1,lh2,ce
      common /VSCOIL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
      integer*2 ilh(3)
      double precision lh(3)
      equivalence (ilh,ilh0),(lh,lh0)

      integer*2 ix,jx
*************************************************************************
      ix=i+1
      jx=j+1

      varray(ix)=varray(jx)
      farray(ix)=farray(jx)
      lh(ix)=lh(jx)
      ilh(ix)=ilh(jx)

      return
      end
*
```

```
e! sys final/f2for/subvx for##
      subroutine SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
     &  htanaf,index,iftyp,nerr)
*****************************************************************************
      implicit double precision (a-z)

      integer*2 nct,ncc,index,iftyp,nerr
      double precision zt(25),vc0t(6),zc(25),vc0c(6),sinaf,cosaf,tanaf,
     &  secaf,htanaf

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision v0,v1,v2,f0,f1,f2,f,eps
      common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps
      double precision varray(3),farray(3)
      equivalence (v0,varray),(f0,farray)

      integer*2 ilh0,ilh1,ilh2,il
      double precision lh0,lh1,lh2,ce
      common /VSCOIL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
      integer*2 ilh(3)
      double precision lh(3)
      equivalence (ilh,ilh0),(lh,lh0)

      integer*2 ix
*****************************************************************************
      ix=index+1
      zt(2)=varray(ix)
      call CALC2(nct,zt,vc0t,sinaf,cosaf,tanaf,secaf,htanaf,ione,nerr)
      goto (110,120,130,130),iftyp
  110 continue
      fval=zt(1)
      goto 200
  120 continue
      fval=zt(2)
      goto 200
  130 continue
      fval=zt(2)-zt(1)*tanaf
      lh(ix)=fval
      if (iftyp eq 3) goto 200
```

```
      ce=WGTH(fval,ncc,zc,vc0c)
      call CERISR(ce,zt,fval)
      ilh(ix)=il
  200 continue
      farray(ix)=fval-f
      return
      end
*
```

```
et sys final/t2for/wgth for$$
      function WGTH(lenh,nc,z,vc0)
*****************************************************************************
      implicit double precision (a-z)

      integer*2 nc
      double precision wgth,lenh,z(25),vc0(6)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      integer*2 ilh0,ilh1,ilh2,il
      double precision lh0,lh1,lh2,ce
      common /VSCOIL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
      integer*2 ilh(3)
      double precision lh(3)
      equivalence (ilh,ilh0),(lh,lh0)

      integer*2 i,is,in
*****************************************************************************
      len1=zero
      is=3*nc
      in=2*nc
      do 100 i=1,nc
      len1=len1+z(is)
      if (i lt nc  and  lenh gt len1) goto 20
      wgth=vc0(in)+(lenh-len1+z(is))*z(is+1)
      il=i
      goto 110
   20 continue
      is=is-3
      in=in-2
  100 continue
  110 continue
      return
      end
*
```

```
e! sys final/f2for/cerisr for##
      subroutine CERISR(ce,zf,ybb)
*************************************************************************
      implicit double precision (a-z)

      double precision ce,zf(25),ybb

      integer*2 ileg,isf,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,fa,fb
      common /VCLOB/ ileg,isf,nca,ncb,z,cz,cx,d,fa,fb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
```

```
      integer*2 iiant
      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,uf,st,ykt,zkt,eex,eez,eey,ybuoy
      common /VCSSHP/ a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,uf,st,ykt,zkt,eex,eez,eey,ybuoy,iiant
*******************************************************************************
      tana7=zi(iiant)+(ce+c3)/h
      call TRISR
      ybb=ykt+zi(12)+y4
      return
      end
*
```

```
e! sys final/t2for/secvit for##
      subroutine SECVIT(nct,zt,vc0t,ncc,zc,vc0c,
     &  sinaf,cosaf,tanaf,secaf,htanaf,iftyp,ifail)
*************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 nct,ncc,iftyp,ifail
      double precision zt(25),vc0t(6),zc(25),vc0c(6),
     &  sinaf,cosaf,tanaf,secaf,htanaf

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      integer*2 ivs
      double precision v0,v1,v2,f0,f1,f2,f,eps
      common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
      double precision varray(3),farray(3)
      equivalence (v0,varray),(f0,farray)

      integer*2 nit,nerr,nr

      equivalence (nerr,nr)
*************************************************************************
*     write(10,*) 'SECVIT',v0,f0,v1,f1,f,iftyp

      ifail=0
      nit=1

 1000 continue
      v2=v1-f1*(v1-v0)/(f1-f0)
 1010 continue
      call SUBVX(nct,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
     &  htanaf,2,iftyp,nerr)
*     write(10,*) 'CALC2',nit,v2,f2,nerr
      if (nerr eq 0) goto 1200
      nr=nerr-nerr/3
      v2=half*(v1+vc0t(nr)+htanaf)
      goto 1010

 1200 continue
```

```
      if (dabs(f2) lt eps) goto 2000
      if (nit gt 50) goto 1900
      call SHIFT(0,1)
      call SHIFT(1,2)
      nit=nit+1
      goto 1000

 1900 continue
      ifail=1
 2000 continue
      return
      end
*
```

```
et sys final/f2for/stefab for##
      subroutine STEFAB
*************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      integer*2 uz1,uz2,uz3,uz4,uz5
      equivalence (uz(1),uz1),(uz(2),uz2),(uz(3),uz3),(uz(4),uz4),
```

```
&  (uz(5),uz5)

 double precision pi,halfpi,degrad,raddeg,zero,one,half
 integer*2 izero,ione,itwo
 common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
&  izero,ione,itwo

 double precision delyk,twod,halfd,dsq
 common /VANCH/ delyk,twod,halfd,dsq

 double precision snphih,csphih,snafh,csafh,tnafh,scafh,dsnph
 common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh,dsnph

 double precision epsy,gamma,se
 integer*2 ia,ib,ie
 common /VCMPD/ epsy,gamma,se,ia,ib,ie

 double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
&  eex0,eez0,eey0,a0,b0,phia0,phib0
 integer*2 icase
 common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
&  eex0,eez0,eey0,a0,b0,phia0,phib0,
&  icase

 double precision a,b,snphi,tnafa,tnafb,
&  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
 common /VCSSHP/ a,b,snphi,tnafa,tnafb,
&  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy

 integer*2 ivs
 double precision v0,v1,v2,f0,f1,f2,f,eps
 common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs

 double precision xred
 integer*2 isidf,nerra,nerrb
 common /VSTAB/ xred,isidf,nerra,nerrb

 integer*2 nit,ipoint,i,ifail,ivsq,iswl

 double precision jac(2,2)
 equivalence (jac(1,1),j11),(jac(1,2),j12),(jac(2,1),j21),
&  (jac(2,2),j22)
```

```
      double precision zz(7)
      equivalence
     &  (zz(1),asav,delamx,delanw,anew),
     &  (zz(2),aold,dela,chnga,anedge),
     &  (zz(3),bold,delb,chngb,bnedge),(zz(4),y1,dcoeff),(zz(5),y2),
     &  (zz(6),ysqo,y1x,aedge),(zz(7),y2x,bedge)
      double precision delx(2)
      equivalence (dela,delx)

      equivalence (cosmx,scafz,halfdd,sl,bsav,delmx,delbmx,delbnw,bnew),
     &  (csphin,hddcsp,xmid,temp1,ysqsav,del),(temp,fact,rat)

      fun1(arga,argb,argd)=arga*arga+dsq-argb*argb+argd*(snphih+snphih)
     &  *arga
*************************************************************************
* Set constant terms
*************************************************************************
      epsysq=epsy*epsy
      zp9=0.9d0

*************************************************************************
* Set iteration switch for subroutine XSECV to zero
*************************************************************************
      ivs=0

*************************************************************************
* Compute upper bounds for a,b, they might not be least upper bounds
*************************************************************************
      amx=one
      bmx=one
      if (hafh.le.zero) goto 120
      cosmx=dmax1(csafh,one/SECNT(cz))
      if (cz.lt.zero) amx=cosmx
      if (cz.gt.zero) bmx=cosmx
  120 continue
      amx=amx*sa
      bmx=bmx*sb

*************************************************************************
* Compute initial guess for (a,b) and assign to anew,bnew
* Set (a,b) equal to nearest asymptotic point
*************************************************************************
      scafz=SECNT(cz)
```

```
      csphin=csafh*(snphih+cz*inafh)/scafz
      halfdd=halfd*scafz
      hddcsp=halfdd*csphin
      temp=hddcsp*hddcsp-halfdd*halfdd
      s1=dmin1(dsqrt(sa*sa+temp)+hddcsp,dsqrt(sb*sb+temp)-hddcsp)

      do 300 i=1,7
      zz(i)=z(i)
  300 continue
      ha=h
      s1a=s1
      w1a=(ca+cb)/s1
      c1a=c3
      s2a=s4
      w2a=w4
      call VCRIT0(itwo,za,vc0a)

      f=do
      call XSECV(2,za,vc0a,s1a+s2a,vc0a(1),nwa,ncb,zb,vc0b,
     &  snafh,csafh,inafh,scafh,2,ifail)

      xmid=za(1)
      temp=xmid*xmid+halfd*halfd
      temp1=xmid+dsnph
      anew=dsqrt(temp-temp1)
      bnew=dsqrt(temp+temp1)
      a=half*(anew+bnew-dsnph)
      b=a+dsnph

      do 500 i=1,7
      z(i)=zz(i)
  500 continue
      call VCRIT0(nca,za,vc0a)

*****************************************************************************
* Beginning of code for Steffensen iteration
*****************************************************************************
* Test new point (a,b) for validity via subroutine CALC3 and adjust
* if necessary, while generating error vector (y1,y2)
* Point (a,b) lies within hyperbolic region, but value of a or b
* may be too large   Point (aold,bold) has passed test with CALC3
*****************************************************************************
      nit=0
```

```
      isq=0
 1000 continue
*     write(10,*)
*     write(10,*) 'ITER',nit
      isw1=0
 1010 continue
      isidf=1
      call CALC3(a,b,y1,y2)
      ysq=y1*y1+y2*y2
      if (nerra+nerrb eq 0  and  (nit eq 0  or  isw1 eq 1
     &   or  ysq lt ysqo*1.d0)) goto 1200
      if (nit eq 0  or  isw1 eq 1) goto 1100
      a=half*(a+aold)
      b=half*(b+bold)
      goto 1010
 1100 continue
      temp=dmax1(xred+xred,half*(halfd*(one-snph)-a))
      a=a+temp
      b=b+temp
      goto 1010

 1200 continue
      if (nit eq 0) goto 2400
      if (isw1 eq 0  or  ysq lt ysqsav) goto 1250
      a=aav
      b=bsav
      ysq=ysqsav
      isw1=0
      goto 1300
 1250 continue
      if (isw1 eq 0  and  ysq ge zp9*ysqo) isq=isq+1

 1300 continue
      if (isq le 4) goto 1400
      asav=a
      bsav=b
      ysqsav=ysq
      a=half*(a+b-dsnph)
      b=a+dsnph
      isw1=1
      isq=0
      goto 1010
```

```
*************************************************************************
* Finished if error vector is sufficiently small or current test point
* is sufficiently close to previous test point
*************************************************************************
 1400 continue
      if (ysq lt epsysq) goto 5000
      if (ysq lt epsysq*1 0d8  and
     &  dabs(one-a/aold)+dabs(one-b/bold) lt 1 0d-8) goto 5000

*************************************************************************
* Compute deltas for Jacobian matrix estimate
*************************************************************************
      if (nit eq 1  or  iswi eq 1) goto 1500
      dela=(j11*y1+j12*y2)*dsqrt(j21*j21+j22*j22)/detj
      delb=(j21*y1+j22*y2)*dsqrt(j11*j11+j12*j12)/detj
      goto 2000
 1500 continue
      dela=dsqrt(half*ysq)
      delb=dela

*************************************************************************
* Adjust deltas as necessary
*************************************************************************
 2000 continue
*     write(10,*) 'init del',dela,delb
      rat=dmax1(one,epsy/dabs(dela),epsy/dabs(delb))
      dela=rat*dela
      delb=rat*delb
      delmx=dmin1(dabs(a-b+d),dabs(b-a+d),a+b-d)
      delamx=dmin1(amx-a,delmx)
      delbmx=dmin1(bmx-b,delmx)
      rat=dmin1(one,delamx/dabs(dela),delbmx/dabs(delb))
      if (rat ne one) rat=0 1d0*rat
      dela=dela*rat
      delb=delb*rat

 2005 continue
      delanw=dela
      if (fun1(a+delanw,b,d) gt zero) goto 2110
      call EDGPT(a,b,a+delanw,b,d,aedge,bedge)
      delanw=aedge-a
 2110 continue
      if (fun1(b,a+delanw,-d) gt zero) goto 2120
```

```
      call EDGPT(b,a,b,a+delanw,-d,bedge,aedge)
      delanw=aedge-a
 2120 continue
      delbnw=delb
      if (funl(a,b+delbnw,d) gt zero) goto 2130
      call EDGPT(a,b,a,b+delbnw,d,aedge,bedge)
      delbnw=bedge-b
 2130 continue
      if (funl(b+delbnw,a,-d) gt zero) goto 2140
      call EDGPT(b,a,b+delbnw,a,-d,bedge,aedge)
      delbnw=bedge-b
 2140 continue

      rat=one
      if (dela eq delanw  and  delb eq delbnw) goto 2160
      rat=dmin1(dabs(delanw/dela),dabs(delbnw/delb))*0.1d0
 2160 continue
      dela=rat*dela
      delb=rat*delb
*     write(10,*) 'fin del',dela,delb,rat

*     if (funl(a+dela,b,d) gt zero  and  funl(b,a+dela,-d) gt zero
*    &   and  funl(a,b+delb,d) gt zero  and  funl(b+delb,a,-d) gt zero)
*    &  goto 2190
*     dela=half*dela
*     delb=half*delb
*     goto 2005
*2190 continue

*******************************************************************************
* Estimate Jacobian matrix
*******************************************************************************
 2200 continue
      ipoint=1
 2205 continue
      del=delx(ipoint)
      isidf=1
      if (del lt zero) isidf=2
      if (ipoint eq 2) isidf=3-isidf
      call CALC3(a+(2-ipoint)*dela,b+(ipoint-1)*delb,y1x,y2x)
      if (nerra+nerrb eq zero) goto 2220
      dela=half*dela
      delb=half*delb
```

```
      goto 2200
 2220 continue
      jac(1,ipoint)=(y1x-y1)/del
      jac(2,ipoint)=(y2x-y2)/del
      ipoint=ipoint+1
      if (ipoint le 2) goto 2205
*     write(10,*) 'jac',j11,j12
*     write(10,*) '   ',j21,j22

*************************************************************************
* Invert Jacobian matrix and compute new point (anew,bnew)
*************************************************************************
      detj=j11*j22-j12*j21
      temp=j11
      j11=j22/detj
      j22=temp/detj
      j12= -j12/detj
      j21= -j21/detj
      anew=a-(j11*y1+j12*y2)
      bnew=b-(j21*y1+j22*y2)
*     write(10,*) 'inv',j11,j12
*     write(10,*) '   ',j21,j22

*************************************************************************
* Adjust new point (anew,bnew) as necessary
*************************************************************************
 2400 continue
*     write(10,*) 'init new pt',anew,bnew
      fact=one
      if (dabs(anew-bnew) lt d) goto 2500
      fact=zp9
      dcoeff=d
      if (anew gt bnew) dcoeff= -d
      chnga=anew-a
      chngb=bnew-b
      anew=(chnga*(b-dcoeff)-chngb*a)/(chnga-chngb)
      bnew=anew+dcoeff
 2500 continue
*     write(10,*) anew,bnew
      if (anew+bnew gt d) goto 2600
      fact=zp9
      chnga=anew-a
      chngb=bnew-b
```

```
      anew=(chnga*(d-b)+chngb*a)/(chnga+chngb)
      bnew=d-anew
 2600 continue
*     write(10,*) anew,bnew
      if (fun(anew,bnew,d) gt zero) goto 2800
      fact=zp9
      call EDCPT(a,b,anew,bnew,d,anedge,bnedge)
      anew=anedge
      bnew=bnedge
 2800 continue
*     write(10,*) anew,bnew
      if (fun(bnew,anew,-d) gt zero) goto 2900
      fact=zp9
      call EDCPT(b,a,bnew,anew,-d,bnedge,anedge)
      anew=anedge
      bnew=bnedge
 2900 continue
*     write(10,*) anew,bnew
      anew=a+fact*(anew-a)
      bnew=b+fact*(bnew-b)
*     write(10,*) 'fin new pt  ',anew,bnew,fact

*     if (fun(anew,bnew,d) gt zero  and  fun(bnew,anew,-d) gt zero)
*    &  goto 3000
*     anew=half*(a+anew)
*     bnew=half*(b+bnew)
*     goto 2600
*3000 continue

*******************************************************************************
* Shift values and return to beginning of Steffensen iteration
*******************************************************************************
      aold=a
      bold=b
      ysqo=ysq
      a=anew
      b=bnew
      nit=nit+1
      goto 1000

 5000 continue
      return
      end
```

```
et sys final/t2for/calc3 for##
      subroutine CALC3(a,b,y1,y2)
*************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      double precision a,b,y1,y2

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
```

```
      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
     &  eex0,eez0,eey0,a0,b0,phia0,phib0
      integer*2 icase
      common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
     &  eex0,eez0,eey0,a0,b0,phia0,phib0,
     &  icase

      integer*2 iscopa,iscopb,itana,itanb,ie
      double precision epsy,gamma,se
      common /VCMPD/ epsy,gamma,se,iscopa,iscopb,itana,itanb,ie

      double precision qa,qb,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
      common /VCSSHP/ qa,qb,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy

      double precision v0,v1,v2,f0,f1,f2,f,eps
      common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps

      double precision xred
      integer*2 isidf,nerra,nerrb
      common /VSTAB/ xred,isidf,nerra,nerrb

      integer*2 nsid

      equivalence (sca,scb,tana7)
*****************************************************************************
*     write(10,*) 'CALC3',a,b
      nerra=0
      nerrb=0
      call PHIAB(a,b,a*a-b*b,dsq,twod,phia,phib)
      call HSPLIT
      if (ha gt zero  and  hb gt zero) goto 900
      nerra=1
```

```
      nerrb=1
      goto 5000
  900 continue
      nsid=1
 1000 continue
      goto (1100,1200),isidf
 1100 continue
      sca=SECNT(inafa)
      f=a
      call XSECV(nca,za,vc0a,sa,ca,nwa,ncb,zb,vc0b,
     &  inafa/sca,one/sca,inafa,sca,1,nerra)
      if (nerra eq 0) goto 2000
      goto 5000
 1200 continue
      scb=SECNT(inafb)
      f=b
      call XSECV(ncb,zb,vc0b,sb,cb,nwb,nca,za,vc0a,
     &  inafb/scb,one/scb,inafb,scb,1,nerrb)
      if (nerrb eq 0) goto 2000
      goto 5000
 2000 continue
      if (nsid eq 2) goto 2200
      nsid=2
      isidf=3-isidf
      goto 1000
 2200 continue
      ianal=(c3+ha*za(ianal)+hb*zb(ianb))/h
      call TRISR
      y1=half*(ya+yb)+y4-do
      y2=yb-ya-delyk
      call X4CALC
 5000 continue
*     write(10,*) 'END CALC3',nerra,nerrb,y1,y2
*     write(10,*) va,ya,vb,yb
*     write(10,*)
      return
      end
*
```

```
et sys final/12for/edgpt for##
      subroutine EDGPT(a,b,aa,bb,d,x,y)
*************************************************************************
      implicit double precision (a-z)

      double precision a,b,aa,bb,d,x,y

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,qd,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,qd,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),s1a),(za(4),w1a),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
```

```
      double precision pi ,halfpi ,degrad ,raddeg ,zero ,one ,half
      integer*2 izero ,ione ,itwo
      common /VCONST/ pi ,halfpi ,degrad ,raddeg ,zero ,one ,half ,
     &  izero ,ione ,itwo

      double precision delyk ,twod ,halfd ,dsq
      common /VANCH/ delyk ,twod ,halfd ,dsq

      double precision snphih ,csphih ,snafh ,csafh ,tnafh ,scafh
      common /VHDIR/ snphih ,csphih ,snafh ,csafh ,tnafh ,scafh

      integer*2 iscopa ,iscopb ,itana ,itanb ,ie
      double precision epsy ,gamma ,se
      common /VCMPD/ epsy ,gamma ,se ,iscopa ,iscopb ,itana ,itanb ,ie

      integer*2 itant
      double precision qa ,qb ,snphi ,tnafa ,tnafb ,
     &  seca7 ,seca8 ,ut ,st ,ykt ,zkt ,eex ,eez ,eey ,ybuoy
      common /VCSSHP/ qa ,qb ,snphi ,tnafa ,tnafb ,
     &  seca7 ,seca8 ,ut ,st ,ykt ,zkt ,eex ,eez ,eey ,ybuoy ,itant

      equivalence (dela ,ta) ,(delb ,root1 ,tb) ,(delasq ,coeff2 ,temp ,gamma) ,
     &  (cp ,coeff1 ,eex) ,(coeff0 ,discr ,eez) ,(slope ,eey) ,(dsnph ,rtot)
*************************************************************************
      dsnph=d*snphih
      if (a ne aa) goto 100
      x=a
      y=dsqrt(a*(a+dsnph+dsnph)+dsq)
      goto 1000
  100 continue
      dela=aa-a
      delasq=dela*dela
      delb=bb-b
      cp=a*bb-aa*b
      coeff0=dsq*delasq-cp*cp
      coeff1=cp*delb+dsnph*delasq
      slope=delb/dela
      if (dabs(one-dabs(slope)) gt 1 0d-6)  goto 200
      x= -coeff0/(coeff1+coeff1)
      goto 500
  200 continue
      coeff2=delasq-delb*delb
      discr=dsqrt(coeff1*coeff1-coeff2*coeff0)
```

```
      root1=(-coeff1-discr)/coeff2
      x=(-coeff1+discr)/coeff2
      if (root1 lt x) goto 220
      temp=x
      x=root1
      root1=temp
  220 continue
      if ((a lt aa  and  root1 gt a)  or  (aa lt a  and  x ge a))
     &   x=root1
  500 continue
      if ((x-a)*(x-aa) ge zero) x=aa
      y=b+(x-a)*slope
 1000 continue
      return
      end
*
```

END

DATE
FILMED

5-83

DTIC